GILDA Praticals
Data management system

GILDA Tutors
INFN Catania

EGEE Tutorial

Roma 03.November.2005

FireMan Catalog commands

Browsing the contents of a directory

`glite-catalog-ls [-h] [-q] [-s` *`SERVICE`*`] [-V] [-v] [-c] [-d] [-g] [-l] [-R] [-S] [-t] [-u] [-x]` *`LFN...`*

where LFN (Logical File Name) is the absolute path of the file/directory to list

Main Options:

- **`-l`** request long output
- **`-v`** increase the verbosity level
- **`-R`** request recursive listing
- **`-S`** sort by size
- **`-t`** sort by modification time
- **`-g`** print GUIDs in the long listing
- **`-h`** print a short help
- **`-s SERVICE`** specifies the service endpoint to use
- **`-d`** list the directory node itself instead of its contents

FireMan: glite-catalog-ls example

Example: `$glite-catalog-ls -l /tcaland`

```
-pdrwl-gs--r-l-g---------     792  2005-10-20 12:52:20
   /tcaland/helloworld.jdl
-pdrwl-gs--r-l-g---------   25890k 2005-09-15 11:53:44
   /tcaland/madagascar.mov
-pdrwl-gs--r-l-g---------     237  2005-10-20 12:52:37
   /tcaland/runshell.jdl
-pdrwl-gs--r-l-g---------      38  2005-10-20 12:51:04
   /tcaland/runshell.sh
```

Meaning of the flags:

- The first letter shows the type of the entry: '**-**' for regular files, '**d**' for directories, '**l**' for symbolic links and '**v**' for virtual directories.
- It follows 3 series of 8 flags, respectively for the owner, the group and others: **p** indicates the permission to change attribute, while **d** gives rights to delete the entry. It follows permissions to **r**ead, **w**rite, **l**ist, e**x**ecute. The last two flags are reserved for metadata use, and so are currently unused. They will give the rights to get or set the metadata.

FireMan: Handling with directories

Directory creation

```
glite-catalog-mkdir [-h][-q][-s SERVICE][-V][-v]
  [-p][-c] DIR...
```

Where **DIR** is the name of the directory to create (in LFN format)

`-p` create missing parent directory

`-c` copy the permission of the parent directory

Example: `glite-catalog-mkdir /tcaland2`

Deleting a directory

```
glite-catalog-rmdir [-h][-q][-s SERVICE][-V][-v]
  DIR...
```

The specified directory must be empty or the command will fail

Example: `glite-catalog-rmdir /tcaland2`

FireMan: Detailed info on a entry

`glite-catalog-stat [-h] [-q] [-s SERVICE] [-V] [-v] [-r] LFN...`

Lists all information the file catalog has about a file or directory. It includes LFN, GUID, owner/group with basic permission, list of ACLs (if any), SURL for every replica.

Example: `glite-catalog-stat /tcaland/runshell.sh`

```
LFN:          /tcaland/runshell.sh
  Created:      2005-10-20 12:51:04.000
  Modified:     2005-10-20 12:51:04.000
  Size:         38
  Type:         File
  Expires:      Never
GUID:           004c3024-7697-1357-9128-c1ced08dbeef
  Created:      2005-10-20 12:51:08.000
  Modified:     2005-10-20 12:51:08.000
  Size:         38
  Checksum:     00000000
  Status:       0
User:           /C=IT/O=INFN/OU=Personal Certificate/L=Catania/CN=Tony Calanducci
Group:          egee-group
User rights:    pdrwl-gs
Group rights:   --r-l-g-
Other rights:   --------
Replicas:
  srm://aliserv6.ct.infn.it:8443/srm/managerv1?SFN=/dpm/ct.infn.it/home/gilda/tcaland/runshell.sh
        Master replica
        Created:        2005-10-20 12:51:04.000
        Modified:       2005-10-20 12:51:04.000
```

FireMan: looking for files

Locate files matching a specified pattern in the specified directory

```
glite-catalog-find [-h] [-q] [-s SERVICE] [-V] [-v] [-n
  LIMIT] PATTERN DIR...
```

Where

- `-n LIMIT` return no more than *LIMIT* results. The default is 100
- ***PATTERN*** is the file name pattern to look for. Recognized wildcards are **?** matching any character, and ***** matching any string
- ***DIR*** Name of the directories where to look for files matching *PATTERN*.

- Example: `glite-catalog-find *world.jdl /tcaland`

```
/tcaland/helloworld.jdl
```

Upload a local file to the Storage Element

glite-put <localfilename> <lfn>[-m <mode>][-c <config>]

where <localfilename> is the name of the local file you want to upload and <lfn> is the logical file name you want to assign to that file.

- **glite-put** is part of the glite-io client CL tools. It will contact the local glite-IO server to accomplish its works. It also interacts with FireMan interfaces to register the new entry into the catalog.

Example: **glite-put hostname.jdl /tcaland/myjob.jdl**

```
[glite_put] Total 0.00 MB       |====================| 100.00 % [0.0 Mb/s]

Transfer Completed:

  LFN                      : /tcaland/myjob.jdl
  GUID                     : 002b06c4-795e-1357-a628-c1ced08dbeef
  SURL                     :
   srm://aliserv6.ct.infn.it:8443/srm/managerv1?SFN=/dpm/ct.infn.it/home/gilda
   /tcaland/myjob.jdl
  Data Written [bytes]     : 202
  Eff.Transfer Rate[Mb/s]  : 0.000076
```

Retrieve and delete a file from the SE

`glite-get <lfn> <localfilename> [-c <config>]`

Copies the given file from the local Storage Element to the given local file.

- <lfn> is the logical file name of the file you want to download and <localfilename> is the name of the destination local file
- -c <config> use an alternativi config file, overriding the glite-io-client configuration

Example: `glite-get /tcaland/myjob.jdl hostname2.jdl`

Transfer Completed:

LFN : /tcaland/myjob.jdl
GUID : 002b06c4-795e-1357-a628-c1ced08dbeef
SURL : srm://aliserv6.ct.infn.it:8443/srm/managerv1?SFN=/dpm/ct.infn.it/home/gilda/tcaland/myjob.jdl
Data Written [bytes] : 0
Eff.Transfer Rate[Mb/s] : 0.000000

`glite-rm <lfn> [-c <config>]`

Removes the given file from your local Storage Element and delete the corresponding entry in the File Catalog

Example: `glite-rm /tcaland/myjob.jdl`

Unlink Completed:

File : /tcaland/myjob.jdl
Time [s] : 4.471000

Hands-on session

Exercise:

- Create a local text file containing whatever information you like
- Give a look inside the file catalog: verify a /rome directory exists
- Create under the /roma a folder with your surname
- Upload the previous created file into your own directory
- Give a look to the details of the just uploaded file
- Download your text file using a different local name
- Remove it from the SE, unregister the entry in catalog and verify its deletion

Fireman commands (I)

Summary of the Fireman Catalog commands

glite-catalog-ls	List file/directory entries in a directory
glite-catalog-mkdir	Create a directory
glite-catalog-mv	Rename a file/directory
glite-catalog-rm glite-catalog-rmdir	Remove a file/directory
glite-catalog-getreplica	Get all replicas associated with a file/GUID
glite-catalog-touch glite-catalog-create	Create a new entry in the catalog/update the modification time
glite-catalog-find	Find entries based on their name pattern
glite-seindex-list	List all SEs having a replica of the given files

Fireman commands (II)

Summary of the Fireman Catalog commands

glite-catalog-chmod glite-catalog-setacl glite-catalog-setdefacl glite-catalog-setdefperm	Change access mode of the Fireman file/directory. Set the ACL, the default ACL and the default permission
glite-catalog-stat glite-catalog-getguid	List the details of a file – all attributes, replicas. Or just the associated GUID.
glite-catalog-setattr glite-catalog-getattr glite-catalog-setschema	Set/get metadata attribute and set the metadata schema of a given directory
glite-catalog-getacl glite-catalog-getdefacl	Get file/directory access control lists and default ACL
glite-catalog-symlink	Make a symbolic link to a file. Directory symlinks are not supported by design.

Practicals on LFC and lcg-utils

Giuseppe Platania

(Thanks to Tony Calanducci)

Check your environment

- **By default the following environment variables are set by UI's profile:**
  - **LCG_CATALOG_TYPE=lfc**
  - **LFC_HOST=lfc-gilda.ct.infn.it**
- **Ensure you have created a proxy certificate and it is still valid. If not create it by:**
  - `grid-proxy-init`
  - Remember: The Passphrase is ROMA

LFC Catalog commands

Listing the entries of a LFC directory

lfc-ls [-cdiLlRTu] [--comment] path…

where *path* specifies the LFC pathname (mandatory)

- Remember that **LFC has a directory tree structure**
- /grid/<VO_name>/<you create it>

- All members of a given VO have read-write permissions under their directory
- ***-l*** (it is a lowercase "L") outputs long listing
- ***-R*** lists the contents of directories recursively (don't use it so often!)
- You can set LFC_HOME to use relative paths
  LFC_HOME=/grid/gilda/myDir ➔ */grid/gilda/myDir/myFile* becomes *myFile*

lfc-ls examples

```
$ lfc-ls -l /grid/gilda
```

```
...
-rw-rw-r--   1 4401     4400                0 Jun 21 09:40 tutor02-rel-pippo-pluto
-rw-rw-r--   1 4401     4400                0 Jun 21 09:39 tutor14
-rw-rw-r--   1 4401     4400                0 Jun 21 09:40 tutor16-mytxt
-rw-rw-r--   1 4401     4400                0 Jun 21 09:32 unitprot-ibcp02
-rw-rw-r--   1 4401     4400                0 Jun 21 09:36 uploadfile
-rw-rw-r--   1 4401     4400                0 Jun 21 09:36 uploadfilelfn
-rw-rw-r--   1 4401     4400                0 Jun 21 09:38 user.example
-rw-rw-r--   1 4401     4400                0 Jun 21 09:38 user.example2
-rw-rw-r--   1 4401     4400                0 Jun 21 09:40 valencia15.ejemplo
-rw-rw-r--   1 4401     4400                0 Jun 21 09:40 valencia15.example
...
```

```
$ export LFC_HOME=/grid/gilda/
$ lfc-ls -l user.example
```

```
-rw-rw-r--   1 4401     4400                0 Jun 21 09:38 /grid/gilda/user.example
```

LFC Catalog commands

Creating a symbolic link

lfc-ln -s file linkname

lfc-ln -s directory linkname

Create a link to the specified *file* or *directory* with *linkname*

– *Example:*

$ lfc-ln -s /grid/gilda/user.example /grid/gilda/roma/linkToUser.ex

Let's check the link using lfc-ls with long listing (-l)

$ lfc-ls -l /grid/gilda/roma

```
lrwxrwxrwx   1 4404     4400   0 Jul 17 12:06 linkToUser.ex ->
   /grid/gilda/user.example
```

LFC Catalog commands

Creating directories in the LFC

lfc-mkdir [-m mode] [-p] path...

- Where *path* specifies the LFC pathname
- Remember that while registering a new file (using lcg-cr, for example) the corresponding destination directory must be created in the catalog before

- Examples:

  $ lfc-mkdir /grid/gilda/Examples

  You can just check the directory with:

  $ lfc-ls -l /grid/gilda

LFC Catalog commands

Adding/deleting metadata information

lfc-setcomment path comment

lfc-delcomment path

lfc-setcomment adds/replaces a *comment* associated with a file/directory in the LFC Catalog

lfc-delcomment deletes a comment previously added

- Example:

  lfc-setcomment /grid/gilda/user.example “Hello Roma”

- Check your job with..

  lfc-ls --comment /grid/gilda/user.example

```
/grid/gilda/user.example Hello Roma
```

LFC Catalog commands

- Example:

lfc-delcomment /grid/gilda/user.example

- Check your job with..

lfc-ls –l --comment /grid/gilda/user.example

```
-rw-rw-r--   1 4401     4400            0 Jun 21 09:38 /grid/gilda/user.example
```

Hands-on Session

Exercise No.1:

- Log onto an UI and initialize your proxy credentials if not already done
- verify that the environment variables point to lfc-gilda.ct.infn.it catalog
- have a look inside the catalog
- create a directory with your surname
- put inside the just created dir a link to an existing file
- add a comment to that file and verify it

LFC Catalog commands

Summary of the LFC Catalog commands

lfc-chmod	Change access mode of the LFC file/directory
lfc-chown	Change owner and group of the LFC file-directory
lfc-delcomment	Delete the comment associated with the file/directory
lfc-getacl	Get file/directory access control lists
lfc-ln	Make a symbolic link to a file/directory
lfc-ls	List file/directory entries in a directory
lfc-mkdir	Create a directory
lfc-rename	Rename a file/directory
lfc-rm	Remove a file/directory
lfc-setacl	Set file/directory access control lists
lfc-setcomment	Add/replace a comment

lcg-utils

- The LCG Data Management tools (usually called ***lcg-utils***) allow users to copy files between UI, WN and a SE, to register entries in the File Catalogs and replicate files between SEs.

- It's important that the LCG_GFAL_INFOSYS environment variable is correctly set to the GILDA Information Index (BDII)

  - **echo $LCG_GFAL_INFOSYS**
    grid004.ct.infn.it:2170

lcg-utils: lcg-cr

Upload a file to a SE and register it into the catalog

- lcg-cr -d dest_file | dest_host -l lfn [-g guid] [-l lfn]
  [-v | --verbose] --vo vo src_file

where

- ***dest_host*** is the fully qualified hostname of the destination SE
- ***dest_file*** is a valid SURL (both sfn:// or srm:// format are valid)
- ***guid*** specifies the Grid Unique IDentifier. If this option is not present, a GUID is generated internally
- ***lfn*** specifies the Logical File Name associated with the file
- ***vo*** specifies the Virtual Organization the user belongs to
- ***src_file*** specifies the source file name: the protocol can be *file:///* or *gsiftp:///*

lcg-utils: lcg-cr

- **To discover which SEs the user is allowed to use, remember you can use lcg-infosites command.**

```
lcg-infosites --vo gilda se
```

The output is a list of SEs and related information on available/used space

- **lcg-cr usage example:**

```
$ lcg-cr -v -d grid-se.bio.dist.unige.it -l lfn:/grid/gilda/roma/note.txt --
   vo gilda file:///home/local/note.txt
Using grid catalog type: lfc
Source URL: file:///home/local/note.txt
File size: 51
Destination specified: grid-se.bio.dist.unige.it
Destination URL for copy: gsiftp://grid-
   se.bio.dist.unige.it/flatfiles/SE00/gilda/generated/2005-07-17/file1f0e73d8-7e3f-
   47d1-bc95-c03c92aae569
# streams: 1
Alias registered in Catalog: lfn:/grid/gilda/roma/note.txt
Transfer took 11320 ms
Destination URL registered in Catalog: sfn://grid-
   se.bio.dist.unige.it/flatfiles/SE00/gilda/generated/2005-07-17/file1f0e73d8-7e3f-
   47d1-bc95-c03c92aae569
guid:4c10a8e3-2244-4c38-bc98-ed98ae7cb94e
```

lcg-utils: lcg-aa and lcg-la

Adding an alias for a given GUID

lcg-aa --vo vo guid lfn

where

- ***vo*** specifies the Virtual Organization the user belongs to
- ***guid*** specifies the Grid Unique Identifier of the file you want to add the alias to
- ***lfn*** specifies the new alias

- *Example:*

```
$ lcg-aa --vo gilda guid:4c10a8e3-2244-4c38-bc98-ed98ae7cb94e
  lfn:/grid/gilda/roma/aliasToNote.txt
```

- To check if the previous command was successful, you can use **lcg-la** command to **list the aliases for a given LFN, GUID or SURL**

```
$ lcg-la --vo gilda lfn:/grid/gilda/roma/aliasToNote.txt
lfn:/grid/gilda/roma/note.txt
lfn:/grid/gilda/roma/aliasToNote.txt
```

Hands-on session

Exercise No.2:

- verify that your **LCG_GFAL_INFOSYS** is correctly set up
- create a dummy file
- check the available storage elements
- copy and register the previous created file into your previously created dir
- add an alias to the just uploaded file
- check if the alias was assigned correctly

lcg-utils commands for replicas (I)

Copying a file from one SE to another one and register it in the Catalog

lcg-rep -d dest_file | dest_host [-v | --verbose] --vo vo src_file

where

- ***dest_host*** is the fully qualified hostname of the destination SE
- ***dest_file*** is a valid SURL (both sfn:// or srm:// are valid)
- ***vo*** specifies the Virtual Organization the user belongs to
- ***src_file*** specifies the source file name: the protocol can be LFN, GUID or SURL. An SURL scheme can be sfn: for a classical SE or srm:

```
$ lcg-rep -v -d grid009.ct.infn.it --vo gilda lfn:/grid/gilda/roma/note.txt
Using grid catalog type: lfc
Source URL: lfn:/grid/gilda/roma/note.txt
File size: 51
Destination specified: grid009.ct.infn.it
Source URL for copy: gsiftp://grid-se.bio.dist.unige.it/flatfiles/SE00/gilda/generated/2005-07-17/file1f0e73d8-7e3f-47d1-bc95-c03c92aae569
Destination URL for copy: gsiftp://grid009.ct.infn.it/flatfiles/SE00/gilda/generated/2005-07-17/file4f3b4cb2-b5fe-467e-9a3e-1ef602465a17
# streams: 1
Transfer took 2410 ms
Destination URL registered in LRC: sfn://grid009.ct.infn.it/flatfiles/SE00/gilda/generated/2005-07-17/file4f3b4cb2-b5fe-467e-9a3e-1ef602465a17
```

lcg-utils commands for replicas (II)

Listing of replicas for a given LFN, GUID or SURL

lcg-lr --vo vo file

where

- ***vo*** specifies the Virtual Organization the user belongs to
- **file** specifies the Logical File Name, the Grid Unique IDentifier or the Site URL. An SURL scheme can be sfn: for a classical SE or srm:

- **Example:**

$ lcg-lr --vo gilda lfn:/grid/gilda/roma/note.txt

```
sfn://grid-se.bio.dist.unige.it/flatfiles/SE00/gilda/generated/2005-
   07-17/file1f0e73d8-7e3f-47d1-bc95-c03c92aae569
sfn://grid009.ct.infn.it/flatfiles/SE00/gilda/generated/2005-07-
   17/file4f3b4cb2-b5fe-467e-9a3e-1ef602465a17
```

or we got the same output using its GUID

$ lcg-lr --vo gilda guid:4c10a8e3-2244-4c38-bc98-ed98ae7cb94e

lcg-utils commands for replicas (III)

Deleting replicas

- **lcg-del [-a] [-s se] [-v | --verbose] --vo vo file**

where

- ***a*** is used to delete all replicas of the given file
- ***se*** specifies the SE from which you want to remove the replica
- ***vo*** specifies the Virtual Organization the user belongs to
- ***file*** specifies the Logical File Name, the Grid Unique IDentifier or the Site URL. An SURL scheme can be sfn: for a classical SE or srm:.

Example:

- delete one replica

```
$ lcg-del --vo gilda -s grid009.ct.infn.it
  lfn:/grid/gilda/roma/note.txt
```

- delete all the replicas

```
$ lcg-del -a --vo gilda lfn:/grid/gilda/roma/note.txt
```

- let's check if the previous command was successful

```
$ lcg-lr --vo gilda lfn:/grid/gilda/roma/note.txt
```

lcg_lr: No such file or directory

- or by `lfs-ls /grid/gilda/roma` (you will not see anymore note.txt and its alias)

lcg-utils: lcg-cp

Downloading a Grid file in a SE to a local destination

lcg-cp [-v | --verbose] --vo vo src_file dest_file

where

- ***vo*** specifies the Virtual Organization the user belongs to
- ***src_file*** specifies the source file name: the protocol can be LFN, GUID, SURL or local file. An SURL scheme can be sfn: for a classical SE or srm:
- ***dest_file*** specifies the destination. The protocol can be file:/// or gsiftp:///

Example:

```
$ lcg-cp --vo gilda lfn:/grid/gilda/roma/note2.txt
  file:/home/local/note2.txt
Source URL: lfn:/grid/gilda/roma/note2.txt
File size: 51
Source URL for copy:
gsiftp://gilda-se-01.pd.infn.it/shared/gilda/generated/2005-07-
  17/file06c3b28c-465f-489c-be3c-b68728e1ca16
Destination URL: file:/home/local/note2.txt
# streams: 1
Transfer took 1060 ms
```

Hands-on session

Exercise No.3:

- Create two replicas of the file you previously uploaded (you could also use the alias to point it out)
- Check if the operation was successful
- Download the file back in your UI
- Delete just one replica and verify that
- Delete all the replicas and verify that
- Verify if the entry is still into the catalog

Final exercise

- **GOAL:**

  Submit a job that does data management: it will retrieve a file previously registered into the catalog.
- **Steps to follow up:**
  - Create a new file in your UI and put some data into it
  - Choose a SE to upload the file to (hint: use `lcg-infosites`)

    and use the appropriate command to accomplish at this operation (`lcg-cr -v -vo gilda -l lfn:/grid/gilda/roma/<choose an lfn> -d <an SE host> file:`pwd`/<your new file>`)
  - create a script.sh file with the following content:

```
#!/bin/sh
/bin/hostname
#Change the LFN_NAME to download from the Catalog.
echo "Start to download.."
lcg-cp --vo gilda lfn:/grid/gilda/roma/<lfn you choose> file:`pwd`/output.dat
echo "Done.."
```

Final exercise (II)

- **Create the JobWithData.jdl:**

```
Type = "job";
JobType = "Normal";

Executable = "/bin/sh";
Arguments = "script.sh";

VirtualOrganisation = "gilda";

StdOutput = "std.out";
StdError = "std.err";

InputSandbox = {"script.sh"};
OutputSandbox = {"std.out","std.err","output.dat"};
```

- Submit it to the grid
- Retrieve the output and verify the content of output.dat

Summary of lcg-utils commands

Replica Management

lcg-cp	Copies a grid file to a local destination
lcg-cr	Copies a file to a SE and registers the file in the catalog
lcg-del	Delete one file
lcg-rep	Replication between SEs and registration of the replica
lcg-gt	Gets the TURL for a given SURL and transfer protocol
lcg-sd	Sets file status to “Done” for a given SURL in a SRM request

File Catalog Interaction

lcg-aa	Add an alias in LFC for a given GUID
lcg-ra	Remove an alias in LFC for a given GUID
lcg-rf	Registers in LFC a file placed in a SE
lcg-uf	Unregisters in LFC a file placed in a SE
lcg-la	Lists the alias for a given SURL, GUID or LFN
lcg-lg	Get the GUID for a given LFN or SURL
lcg-lr	Lists the replicas for a given GUID, SURL or LFN

Data Transfers: requirements

- **It is mandatory to store your credentials onto a myproxy server. So please do it using:**
  - **myproxy-init -d -s grid001.ct.infn.it**

    (-d flag is used to bound the proxy with your DN)
- **Check which CHANNELS are available for data transfer on your VO, using:**
  - **glite-transfer-channel-list [-h] [-q] [-s SERVICE] [-V] [-v] CHANNEL...**

Examples: **$ glite-transfer-channel-list**

```
channel1
channel2
channel3
channelCtPd
channelPdCt
```

If you are authorized

$ glite-transfer-channel-list channel1

```
Channel: channel1
Between: ct.infn.it and
cnaf.infn.it
State: Active
Contact: fts-admin@ct.infn.it
Bandwidth: 1001
Nominal throughput: 1000
Number of files: 4, streams: 1
Number of VO shares: 2
VO 'gildav' share is: 50
VO 'gilda' share is: 50
```

Move data between SEs

Submit a data placement job

```
glite-placement-submit [-h] [-q] [-s SERVICE] [-V] [-v] [-g PARAM]
   [-e SOURCE_SE] DEST_SE [LFN...]
```

Where

- DEST_SE is the destination Storage Element
- LFN is one or more Logical File Names which need to be trasferred
- SOURCE_SE is the source Storage Element
- PARAM if you need to pass parameters to the underlying gridftp service

glite-placement-submit is used to create replicas of the input file(s). It also properly updates the SURLs for the file(s) in the File Catalog.

- **Example: $ glite-placement-submit -e ct.infn.it cnaf.infn.it /FPS_Example**

  Enter password:

  6685222f-3b34-11da-8336-f118ddd430f7

  - The prompted password is the MyProxy passphrase.
  - A job ID is returned to monitor and check the result of the transfer job

Control Data Transfer Jobs

Display the status of data transfer jobs

glite-placement-status [-h] [-q] [-s SERVICE] [-V] [-v] [-l] JOBID...

- `JOBID` is the Identifier of a submitted placement job returned by glite-placement-submit.
- `-l` list the status of invidual files that are part of the placement job
- **Example:** **glite-placement-status -l 6685222f-3b34-11da-8336-f118ddd430f7**

```
Active
  Source:      srm://aliserv6.ct.infn.it:8443/srm/managerv1?SFN=/dpm/ct.infn.it/home/gilda/FPS_Example
  Destination: srm://egee016.cnaf.infn.it:8443/srm/managerv1?SFN=/dpm/cnaf.infn.it/home/gilda/FPS_Example
  State:       Active
  Failures:    0
  Reason:      (null)
```

P.S.:Probably because of a bug, in gLite-1.3, when job is done correctly, its status is not correctly shown (however in case of failures you can clearly see them...)

Cancel a previously submitted file placement job

glite-placement-cancel [-h] [-q] [-s SERVICE] [-V] [-v] JOBID...

- **Example: glite-placement-cancel 6685222f-3b34-11da-8336-f118ddd430f7**

Hands-on session

Exercise:

- **Store your credentials into a MyProxy server (grid001.ct.infn.it)**
- **Create a text file, upload/register it into the local SE/File Catalog**
- **Check the details of the uploaded file, observing its SURL (Storage URL)**
- **Choose one of the available destination Storage Element (ct.infn.it, cnaf.infn.it, pd.infn.it) and make a replica of the file previously stored.**
- **Check the status of the placement job**
- **Once completed, verify if the File Catalog shows correctly the SURL of the replica.**

Access the Metadata Catalog

- **Initialize your proxy using requesting VOMS gildav membership:**

  $ voms-proxy-init --voms gildav

- **Start up the Metadata Catalog Client with**

  $ mdclient

- **Once logged in, you can list the available commands, typing help.**

  Connecting to :0...
  Connected to wn1-test.ct.infn.it:8822
  ARDA Metadata Server 0.9.6
  Query> help
  >> >help [topic]<
  >> >Displays help on a command or a topic.<
  >> >Valid topics are: help metadata metadata-optional directory entry group acl index schema sequence user view ticket commands<

- Commands are grouped by topic. You can get the list of valid commands for each topic, typing help [topic]
- **Example**: help entry

MDC directory related commands

- **Browse the contents of a directory**
  - **dir [path]**

  Returns the name of all subdirectories and files in the given *path* or in the current directory if not specified
- **Print the current working directory**
  - **pwd**
- **Change the current working directory**
  - **cd directory**

  Example: cd /gildav/tcaland
- **Directory creation**
  - ***createdir /parentdir/dir***

  Creates the directory *dir* if it does not yet exist but *parentdir* already does

  Example: createdir /gildav/tcaland
- **Directory removal**
  - **rmdir path**

  Removes the directory given by path

MDC: Handling attributes

- **Schema population**
  - **addattr dir attr type**

  Adds a new attribute to the schema of a directory. Type is the name of an SQL datatype which will translated (if necessary) into a data type understood by the back end DB.

  Examples of valid datatypes are `int, float, varchar(n), timestamp, text, numeric(p,s)`

  Examples: addattr /gildav/tcaland MovieTitle varchar(100)
  addattr /gildav/tcaland Runtime int
  addattr /gildav/tcaland PlotOutline text
- **Attribute listing**
  - **Listattr path**

  Returns a list of all attributes of the given file/direcory
- **Attribute Removal**
  - **removeattr dir attribute**

  Removes an attribute from a directory if it is not used by any entry in the directory

MDC: managing entries

- **Entry creation**
  - **addentry entry (attribute value)+**

  Add a new entry and initializes some attributes

  Example: addentry /gildav/tcaland/madagascar.mov MovieTitle Madagascar
- **Setting attribute values**
  - **setattr entry (attribute value)+**

  Sets one or more attributes of an entry to given values

  Example: setattr /gildav/tcaland/madagascar.mov Runtime 86
- **Getting attribute values**
  - **getattr pattern (attribute)+**

  Returns the entries and all the attributes for every file matching pattern

  Example: getattr /gildav/tcaland/*.mov Title
- **Entry deletion**
  - **rm pattern**

  Removes all entries matching pattern

  Example: rm /gildav/tcaland/m*.mov

Metadata Queries

- **find pattern ‘query_condition’**

Returns all entries matching pattern for which query_condition is true

Examples:

find /gildav/tcaland/ ‘Runtime > 80’

find /gildav/tcaland/ ‘like(MovieTitle, “Mad%”)’

find /gildav/tcaland ‘like(MovieTitle, “Mad%”) AND Runtime > 80’

- **selectattr attr... condition**

Returns the values of given attributes for all files matching condition

Example:

```
cd /gildav/tcaland
selectattr .:MovieTitle .:Runtime 'Runtime > 80'
>> >Madagascar<
>> >86<
```

Hands-on Session

Exercise:

- Log into the Metadata Catalog
- Create a directory with your surname into the /roma directory
- Add some attributes (Description (varchar(100), Value int, Comment text) to the directory just created
- Add some entries using as entry name the LFNs you uploaded and registered into the File Catalog during the last hands-on session
- Fill the attribute fields for the inserted entries
- Look for the entry with Value > 50